Scalable Stabilization of Large Scale Discrete–time Linear Systems via the 1–norm

Nikolaos Athanasopoulos [⋆] **Mircea Lazar**

Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands, E-mail: {n.athanasopoulos,m.lazar}@tue.nl

Abstract: This article reconsiders the stabilizing controller synthesis problem for discrete–time linear systems with a focus on systems of large scale. In this case, existing solutions are either not scalable, and thus, not tractable, or conservative. This motivates us to exploit finite–time control Lyapunov functions (CLFs), i.e., a relaxation of the standard CLF concept, to obtain a nonconservative and scalable synthesis method. The main idea is to employ Minkowski functions of a particular family of polytopic sets, which includes the hyper–rhombus induced by the 1–norm, as candidate finite–time CLFs. This choice results in explicit periodic vertex–interpolation control laws, which are globally stabilizing. The vertex–control laws can be computed offline using distributed optimization, in a scalable fashion, while the actual control law comes in an explicit, distributed form. Large scale illustrative examples demonstrate the effectiveness of the proposed approach.

Keywords: Large scale linear systems, Lyapunov methods, scalable synthesis, distributed optimization

1. INTRODUCTION

Stability analysis and controller synthesis for large scale linear systems (Michel and Miller, 1977; Vidyasagar, 1981; Siljak, 1991, 2007) remains an active area of research for more than four decades due to its significance and unresolved open problems. The stabilization problem for large scale systems is difficult since both the controller synthesis and implementation are hampered by the state–space dimension.

Solving the stabilizing controller synthesis problem for a large scale linear system *via* standard Lyapunov methods is not tractable. Moreover, even if a solution could be obtained, its implementation would be hampered by the lack of a distributed structure. The latter issue can be tackled using dissipativity theory to set–up a collection of smaller problems with coupling constraints, which are merged into an overall problem and solved as a single optimization problem, see, e.g., (Langbort et al., 2004; Dekker et al., 2010). Usually, the complexity of the resulting problem is higher than the one of the original synthesis problem. However, if it is feasible and a solution can be computed, the resulting controller architecture allows a distributed implementation. Another approach, also based on Lyapunov methods, concerns the use of vector control Lyapunov functions and corresponding comparison systems, of a particular structure, see, e.g., (Grujic and Siljak, 1973; Siljak, 2007). The problem with this approach lies in the fact that imposing an a priori fixed structure is conservative, unless the system has a specific structure as well. Lately, model predictive control based approaches have also been explored, see, e.g., (Johansson et al., 2006; Summers and Lygeros, 2012) and the references therein. These methods exploit distributed optimization algorithms, such as the ones in (Rantzer, 2009; Boyd et al., 2011), to solve subproblems that share a subset of the independent variables and constraints, in order to converge iteratively to the optimal solution of the model predictive control problem for the full–size system.

This article proposes a different approach to stabilization of large scale linear systems, which is based on a recent relaxation of the standard control Lyapunov function (CLF) notion (Athanasopoulos et al., 2013), i.e., finite–time control Lyapunov functions. Similarly to standard synthesis methods based on polyhedral CLFs, see, e.g., (Gutman and Cwikel, 1986), finite–time polyhedral CLFs can be employed to compute stabilizing *periodic* vertex–control laws, while the actual control law is obtained by periodic interpolation among the vertex–control laws (Athanasopoulos et al., 2013). However, in contrast to the standard CLF case, any choice within the class of Minkowski (or gauge) functions of proper C–sets provides a valid finite–time CLF candidate. This observation gives rise to a very relevant question: which family of proper C–polytopic sets provides vertex–control laws of scalable complexity and, moreover, allows distributed implementation of the resulting interpolation–based control law.

The main contribution of this paper is a formal characterization of a family of proper C–polytopic sets that answers the above question. More precisely, this family consists of the 1–norm hyper–rhombus along with its weighted and possibly nonsymmetric variants. The sets within this family have two appealing characteristics, namely, the number of their vertices scales linearly with respect to the state–space dimension, and every vector that lies in the set can be explicitly described as the convex combination of the extreme points of the set. These properties are exploited to obtain a scalable distributed method for synthesis of the vertex–control laws. Furthermore, an explicit and distributed characterization of the resulting interpolation–based control law is derived. The effectiveness of the developed method for stabilization of large scale linear systems is demonstrated on two challenging case studies.

⋆ Supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement 302345

The remainder of the paper is organized as follows. Section 2 introduces preliminary notions and results that are instrumental in deriving the main results. The problem formulation is stated in Section 3. The main results are reported in Section 4, while the illustrative examples are presented in Section 5. Conclusions are drawn in Section 6.

2. PRELIMINARIES

Let $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{N}$ denote the field of real numbers, the set of non-negative reals and the set of nonnegative integers, respectively. For every $c \in \mathbb{R}$ and $\Pi \subseteq \mathbb{R}$ we define $\Pi_{\geq c} := \{k \in \Pi \mid k \geq c\}$, and similarly $\Pi_{\leq c}$, $\mathbb{R}_\Pi := \Pi$ and $\mathbb{N}_\Pi := \mathbb{N} \cap \Pi$. For a matrix $A \in \mathbb{R}^{n\times m}$, $[A]_{ij}$ denotes the element in the i–th row and j–th column, $[A]_{i:} \in \mathbb{R}^m$ denotes the i–th row and $[A]_{:j} \in \mathbb{R}^n$ denotes the j–th column. Given a vector $x \in \mathbb{R}^n$, $[x]_i \in \mathbb{R}$ denotes the i–th entry of x, while vectors $x^+ \in \mathbb{R}^n$ and $x^- \in \mathbb{R}^n$ are defined as $[x^+]_i := \max\{[x]_i, 0\}$, $i \in \mathbb{N}_{[1,n]}$ and $[x^-]_i := \max\{-[x]_i, 0\}$, $i \in \mathbb{N}_{[1,n]}$, such that

$$x = x^+ - x^-.$$

Given a vector $x \in \mathbb{R}^n$, the vector $\overline{x} \in \mathbb{R}^{2n}_+$ is defined as

$$\overline{x} := [{x^+}^\top \quad {x^-}^\top]^\top,$$

where $(\cdot)^\top$ denotes the transpose operation. The identity matrix is denoted by $I_n \in \mathbb{R}^{n\times n}$. The vector with all of its elements equal to one is denoted by $\mathbf{1}_n \in \mathbb{R}^n$, while the vector with all its elements equal to zero is denoted by $\mathbf{0}_n \in \mathbb{R}^n$. A set $\mathcal{S} \subset \mathbb{R}^n$ is *a proper C–set* if it is compact, convex, and contains the origin in its interior. Given a set $\mathcal{S} \subset \mathbb{R}^n$ and a real scalar $\alpha \in \mathbb{R}$, the set $\alpha\mathcal{S}$ is defined by $\alpha\mathcal{S} := \{\alpha x : x \in \mathcal{S}\}$. A *polyhedron* is the (convex) intersection of a finite number of open and/or closed half–spaces and a *polytope* is a closed and bounded polyhedron. In this article, we employ proper $\mathcal{C}$–polytopic sets, which can be defined either by half–space or vertex representations (Ziegler, 2007). Generically, the half–space representation of an arbitrary proper $\mathcal{C}$–polytopic set corresponds to

$$\mathcal{S} := \{x \in \mathbb{R}^n : Hx \leq \mathbf{1}_p\}, \tag{1}$$

where $H \in \mathbb{R}^{p\times n}$ is a full column–rank matrix and $p \in \mathbb{N}_{\geq n+1}$. Generically, the vertex representation of $\mathcal{S}$ corresponds to

$$\mathcal{S} := \mathrm{convh}(\{v_i\}_{i\in\mathbb{N}_{[1,q]}}), \tag{2}$$

for some $q \in \mathbb{N}_{\geq n+1}$. We define $V := [v_1, v_2, \ldots, v_q] \in \mathbb{R}^{n\times q}$ as the corresponding matrix that has as columns the vertices of $\mathcal{S}$. Note that the matrix V has full row–rank. In this article, we will consider proper $\mathcal{C}$–polytopic sets. The following simple algebraic conditions verify if a vector belongs to a proper $\mathcal{C}$–polytopic set.

Fact 1. Let $\mathcal{S} \subset \mathbb{R}^n$ be a proper $\mathcal{C}$–polytopic set described in the half–space (1) and vertex (2) representation, and a vector $x \in \mathbb{R}^n$. Then, $x \in \mathcal{S}$ if and only if **(i)** $Hx \leq \mathbf{1}_p$, **(ii)** there exists a nonnegative vector $p \in \mathbb{R}^q_+$, such that $x = Vp$ and $\mathbf{1}_q^\top p \leq 1$.

Given a proper C–set $\mathcal{S} \subset \mathbb{R}^n$, the function $\mathrm{gauge}(\mathcal{S}, x) := \inf_\mu\{\mu \ : \ x \in \mu\mathcal{S},\ \mu \geq 0\}$, defined for any $x \in \mathbb{R}^n$, is called the Minkowski function, or gauge function. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ *belongs to class* $\mathcal{K}$ if it is continuous, strictly increasing and $\varphi(0) = 0$.

We consider time–invariant non–autonomous linear systems described by the difference equation

$$x_{t+1} = Ax_t + Bu_t, \quad \forall t \in \mathbb{N}, \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ are the system matrices and the matrix pair (A, B) is stabilizable. A system is considered to be a large scale system if $n \in \mathbb{N}_{\geq 100}$ and $m \in \mathbb{N}_{\geq 10}$. Furthermore, no structure (e.g. diagonal A matrix, sparsity etc.) in system (3) is assumed. However, as it will be demonstrated, the results established in this paper are inherently suitable to large scale interconnected systems. We consider the class of state–feedback control laws $g : \mathbb{R}^n \to \mathbb{R}^m$, such that $g(0) = 0$, which are $\mathcal{K}$–bounded at zero, i.e., for all $x \in \mathbb{R}^n$, there exists a $\varphi \in \mathcal{K}$ such that $\|g(x)\| \leq \varphi(\|x\|)$.

The dynamics of system (3) is denoted with $\Phi(x, u) := Ax + Bu$. Then, the k–th iterated map $\Phi^k(x, g(x))$ of the closed–loop dynamics, $\Phi^k(\cdot, g(\cdot)) : \mathbb{R}^n \to \mathbb{R}^n$ is given by $\Phi^k(x, g(x)) := \Phi(\Phi^{k-1}(x, g(x)), g(\Phi^{k-1}(x, g(x))))$, for any $k \in \mathbb{N}_{\geq 1}$. In what follows, the notions of controlled (k, λ)–contractive sets, introduced in (Athanasopoulos et al., 2013), are recalled.

Definition 2. Given a real scalar $\lambda \in \mathbb{R}_{[0,1]}$ and an integer $k \in \mathbb{N}_{\geq 1}$, the proper $\mathcal{C}$–set $\mathcal{S} \subset \mathbb{R}^n$ is called a controlled (k, λ)–contractive set with respect to system (3) if and only if there exists a state–feedback control law $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ such that $\Phi^k(x, g(x)) \in \lambda\mathcal{S}$.

Next, let the set $\mathcal{S}$ be a controlled $(k, 1)$–contractive proper $\mathcal{C}$–set with respect to (3). Suppose there exists a function $V : \mathbb{R}^n \to \mathbb{R}_+$, functions $\kappa_1, \kappa_2 \in \mathcal{K}$, a real scalar $\rho \in (0, 1)$, an integer $k \in \mathbb{N}_{\geq 1}$, and a state–feedback control law $u = g(x)$, $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$, such that the following inequalities hold:

$$\kappa_1(\|x\|) \leq V(x) \leq \kappa_2(\|x\|), \quad \forall x \in \mathbb{R}^n, \tag{4a}$$

$$V(\Phi^k(x, g(x))) \leq \rho V(x), \quad \forall x \in \mathcal{S}. \tag{4b}$$

Definition 3. A function $V(\cdot)$ that satisfies (4) is called a finite–time control Lyapunov function associated with the $(k, 1)$–contractive proper C–set $\mathcal{S}$. A function $V(\cdot)$ that satisfies (4b) for all $x \in \mathbb{R}^n$ is called a global finite–time control Lyapunov function.

It is worth noting that, for the linear case, if relations (4) hold for a proper set $\mathcal{S} \subset \mathbb{R}^n$, then $V(\cdot)$ is a global finite–time control Lyapunov function. The equivalence between controlled (k, λ)–contractive proper $\mathcal{C}$–sets and finite–time CLFs is formally stated next.

Proposition 4. [Lazar et al. (2013), Athanasopoulos et al. (2013)]. Consider the linear system (3) and a proper $\mathcal{C}$–set $\mathcal{S}$. The following statements are equivalent.

(i) The matrix pair (A, B) is stabilizable.

(ii) Given a scalar $\lambda \in \mathbb{R}_{(0,1)}$, there exists a finite integer $k \in \mathbb{N}$ such that the set $\mathcal{S}$ is a controlled (k, λ)–contractive set with respect to system (3).

(iii) The function $V(x) := \mathrm{gauge}(\mathcal{S}, x)$ is a global finite–time CLF.

(iv) There exists a stabilizing state feedback control law $u := g(x)$, $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ such that the closed–loop system $x_{t+1} = \Phi(x_t, g(x_t))$ is globally $\mathcal{KL}$–stable.

When the controlled (k, λ)–contractive set $\mathcal{S}$ is a polytopic proper $\mathcal{C}$–set, a set–induced state–feedback control law can be established. In specific, a general method for finding stabilizing periodic control laws can be formulated by computing input sequences that drive all trajectories starting from the vertices of the set $\mathcal{S}$ in its interior after k time steps. To this end, consider the controlled (k, λ)–contractive set $\mathcal{S} \subset \mathbb{R}^n$, where

$$\mathcal{S} := \{x \in \mathbb{R}^n : H_0 x \leq \mathbf{1}_p\} = \mathrm{convh}(\{v_0^j\}_{j \in \mathbb{N}_{[1:q]}}), \qquad (5)$$

and the matrix $V_0 \in \mathbb{R}^{n \times q}$, $V_0 := [v_0^1, v_0^2, \ldots, v_0^q]$.

Problem 5. Consider the linear system (3) and the controlled (k, λ)–contractive set $\mathcal{S}$ (5). Solve the following q feasibility problems, for each $l \in \mathbb{N}_{[1,q]}$.

$$\min_{\{u_i^l\}_{i \in \mathbb{N}_{[0,k-1]}}, \{v_i^l\}_{i \in \mathbb{N}_{[1,k]}}, p^l} 0 \qquad (6)$$

subject to

$$v_{i+1}^l = A v_i^l + B u_i^l, \ \forall i \in \mathbb{N}_{[0,k-1]}, \qquad (7a)$$

$$v_k^l = V_0 p^l, \qquad (7b)$$

$$p^l \geq 0, \qquad \mathbf{1}_q^\top p^l \leq \lambda, \ \forall i \in \mathbb{N}_{[1,k]}. \qquad (7c)$$

Next, consider the set of matrices $U_i \in \mathbb{R}^{m \times q}$, $i \in \mathbb{N}_{[0,k-1]}$, $V_i \in \mathbb{R}^{n \times q}$, $i \in \mathbb{N}_{[1,k]}$ that are constructed from the solution of Problem 5, i.e.,

$$[U_i]_{:j} := u_i^j, \quad (i,j) \in \mathbb{N}_{[0,k-1]} \times \mathbb{N}_{[1,q]},$$

$$[V_i]_{:j} := v_i^j, \quad (i,j) \in \mathbb{N}_{[1,k-1]} \times \mathbb{N}_{[1,q]},$$

$$V_k := V_0,$$

and consider the control law

$$\pi(x_t) := U_i \mu_i(x_t) \quad \text{if } t = kM + i, \ M \in \mathbb{N}, \qquad (8)$$

for all $i \in \mathbb{N}_{[0,k-1]}$, $\mu_i(x_t) \in \mathcal{M}_i(x_t)$, where

$$\mathcal{M}_0(x_t) := \{\mu \in \mathbb{R}_+^q : x_t = V_0 \mu, \ \mathbf{1}_q^\top \mu \leq 1\}, \qquad (9)$$

$$\mathcal{M}_i(x_t) :=$$

$$\{\mu \in \mathbb{R}_+^q : V_i \mu = (AV_{i-1} + BU_{i-1}) \mu_{i-1}(x_t), \ \mathbf{1}_q^\top \mu \leq 1\}, \qquad (10)$$

for all $i \in \mathbb{N}_{[1,k-1]}$. The next result establishes that the *periodic vertex–interpolation control law* (8)–(10) is stabilizing for system (3).

Proposition 6. Consider system (3), a controlled (k, λ)– contractive set $\mathcal{S}$ as defined in (5) and the corresponding state–feedback control law defined in (8)–(10). The closed–loop system

$$x_{t+1} = A x_t + B \pi(x_t), \quad \forall t \in \mathbb{N} \qquad (11)$$

is $\mathcal{KL}$–stable.

Proof. For any $x_0 \in \mathcal{S}$, the closed–loop system dynamics under the control law (8)–(10) is

$$x_{i+1} = (AV_i + BU_i) \mu_i(x_0), \quad i \in \mathbb{N}_{[0,k-1]}. \qquad (12)$$

Furthermore, from Fact 1 it holds that $x_k \in \lambda \mathcal{S}$, for any selection of $\mu_i(x_0) \in \mathcal{M}_i(x_0)$, $i \in \mathbb{N}_{[0,k-1]}$. Thus, the set $\mathcal{S}$ is (k, λ)–contractive with respect to the closed–loop system, and from (Athanasopoulos et al., 2013, Proposition 2), it follows that the closed–loop system is $\mathcal{KL}$–stable. ■

Remark 7. From Proposition 4, for any given scalar $\lambda \in \mathbb{R}_{[0,1)}$, there always exists a finite integer k, such that conditions (7) hold, for any arbitrary proper $\mathcal{C}$–polytopic set. Thus, the characterization of an arbitrary proper $\mathcal{C}$–polytopic set as a controlled (k, λ)–contractive set is possible and can be performed as follows: Given a desired contraction rate λ, Problem 5 is solved iteratively with increasing k, until a feasible solution is found.

3. PROBLEM FORMULATION

From the established results in Section 2, the computation of stabilizing control laws for system (3) can be performed in two steps. The first step, which concerns the offline *controller synthesis*, involves the selection of a proper $\mathcal{C}$–polytopic set and the construction of the set–valued induced periodic vertex–interpolation control laws (8)–(10). The second step, which concerns the online *controller implementation*, involves the online computation of the stabilizing control law (8)–(10) at each time instant.

In detail, for a chosen proper $\mathcal{C}$–polytopic set, the offline controller synthesis problem requires the solution of Problem 5 with increasing horizon k, until feasibility is achieved. Problem 5 involves the solution of q linear programs, where q is the number of vertices of the chosen proper $\mathcal{C}$–polytopic set. For large scale systems however, the dimensions of the state–space and the input space can render each linear problem difficult to solve. In addition, the number of vertices of the set, thus the number of linear programs to be solved, can increase exponentially with respect to the state–space dimension for an arbitrary selection of the shape of the proper $\mathcal{C}$–polytopic set [1] . Thus, the first problem to be treated concerns the selection of the most suitable family of proper $\mathcal{C}$–polytopic sets whose number of vertices is scalable with respect to the state–space dimension, and the establishment of a tractable method for solving each linear program in Problem 5.

Moreover, since the periodic vertex–interpolation control laws are set–valued, the online implementation requires an admissible selection of vectors $\mu_i(x) \in \mathcal{M}_i(x)$, $i \in \mathbb{N}_{[0,k-1]}$ at every k time instants. The selection of each $\mu_i(x)$, $i \in \mathbb{N}_{[0,k-1]}$, stems from the solution of a linear program which may become intractable for large scale systems and arbitrary proper $\mathcal{C}$–polytopic sets. Thus, the second problem to be treated is to identify a family of sets for which the online controller implementation is explicit and scales linearly with the state and input space dimensions.

In summary, the problem to be investigated can be formulated as follows. *Find a family of proper $\mathcal{C}$–polytopic sets, or equivalently a finite–time CLF parameterization, that render both the controller synthesis and controller implementation problems tractable for large scale linear systems.*

4. MAIN RESULTS

In this section, a tractable solution to both the controller synthesis and implementation problem for large scale systems is provided. First, we introduce an appealing family of proper $\mathcal{C}$–sets, and correspondingly, finite–time CLFs.

4.1 1–*norm family of proper $\mathcal{C}$–polytopic sets*

Consider the diagonal matrices $V^M \in \mathbb{R}^{n \times n}$, $V^m \in \mathbb{R}^{n \times n}$

$$[V^M]_{:i} := [0 \ \cdots v_i^M \cdots 0]^\top, \qquad i \in \mathbb{N}_{[1,n]}, \qquad (13)$$

$$[V^m]_{:i} := [0 \ \cdots - v_i^m \cdots 0]^\top, \qquad i \in \mathbb{N}_{[1,n]}, \qquad (14)$$

where $v_i^M \in \mathbb{R}_{>0}$, $v_i^m \in \mathbb{R}_{>0}$, for all $i \in \mathbb{N}_{[1,n]}$ and the matrix $V \in \mathbb{R}^{2n}$,

$$V := [V^M \quad V^m]. \qquad (15)$$

The vertex representation of this family of sets is

$$\mathcal{S} := \mathrm{convh}(\{[V]_{:i}\}_{[1,2n]}). \qquad (16)$$

The family of sets (13)–(16) possesses several appealing characteristics for the proposed setting. In specific, the number of

[1] For example, the number of vertices of the unit hypercube $\mathcal{B}_\infty := \{x \in \mathbb{R}^n : \|x\|_\infty \leq 1\}$, is 2^n, rendering the construction of the control law (8)–(10) intractable for this particular choice.

vertices of the sets $\mathcal{S}$ is equal to $2n$, i.e., the complexity in their vertex–representation scales linearly with the state–space dimension. Furthermore, for any vector $x \in \mathcal{S}$, the nonnegative weights that describe x as the convex combination of the vertices of $\mathcal{S}$ can be explicitly derived, according to the following fact.

Fact 8. Consider the proper $\mathcal{C}$–polytopic set $\mathcal{S}$ (13)–(15) and a vector $x \in \mathcal{S}$. Then,

$$x := V\mu, \tag{17}$$

where $\mu := [\mu_1^\top \quad \mu_2^\top]^\top$, $\mu_1 \in \mathbb{R}^n$, $\mu_2 \in \mathbb{R}^n$, and

$$[\mu_1]_i := [x^+]_i (v_i^M)^{-1}, \quad i \in \mathbb{N}_{[1,n]}, \tag{18}$$

$$[\mu_2]_i := [x^-]_i (-v_i^m)^{-1}, \quad i \in \mathbb{N}_{[1,n]}. \tag{19}$$

Proof. For all $i \in \mathbb{N}_{[1,n]}$, $[V\mu]_i = v_i^M[\mu_1]_i - v_i^m[\mu_2]_i = [x^+]_i - [x^-]_i = [x]_i$. Thus, $x = V\mu$. ■

Example 9. To illustrate the explicit decomposition of a vector x as a convex combination of the vertices of the family of sets under study, consider the proper $\mathcal{C}$–polytopic set $\mathcal{S}$ of the form (13)–(16), where $v_1^M = 2$, $v_2^M = 1$, $v_1^m = -0.5$ and $v_2^m = -2$,

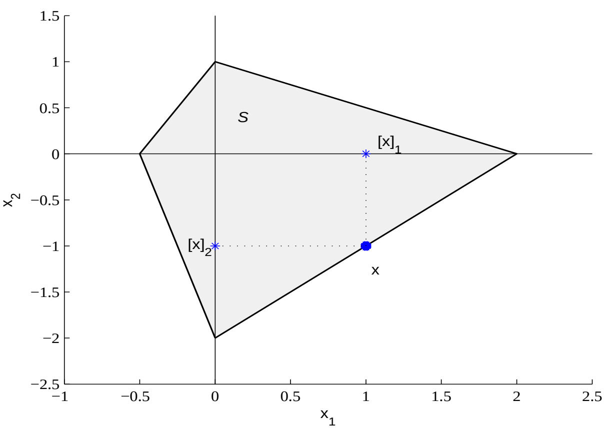

Fig. 1. The set $\mathcal{S}$ (grey), which is a member of the family of sets of the form (13)–(16), and the vector $x \in \mathcal{S}$ (blue dot).

and a vector $x \in \mathcal{S}$, $x = [1 \quad -1]^\top$. From Fact 8 and relations (17)–(19), it follows that $\mu \in \mathbb{R}_+^4$ can be explicitly computed and is equal to

$$\mu = \begin{bmatrix} [x^+]_1 (v_1^M)^{-1} \\ [x^+]_2 (v_2^M)^{-1} \\ [x^-]_1 (v_1^m)^{-1} \\ [x^-]_2 (v_2^m)^{-1} \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{bmatrix}.$$

Indeed,

$$V\mu = \begin{bmatrix} 2 & 0 & -0.5 & 0 \\ 0 & 1 & 0 & -2 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} = x,$$

while $\mathbf{1}_4^\top \mu = 1$.

In order to formulate the controller synthesis problem with the least computational complexity, we consider the simplest member of the polytopic family (13)–(16), i.e., the unit sublevel set of the 1–norm

$$\mathcal{B}_1 := \{x \in \mathbb{R}^n : \|x\|_1 \leq 1\}. \tag{20}$$

The set $\mathcal{B}_1$ is of the form (13)–(16), with $v_i^M = v_i^m = 1$, $i \in \mathbb{N}_{[1,n]}$, while the marix $V \in \mathbb{R}^{2n}$ consisting of its vertices is denoted by $V := [I_n \quad -I_n]$.

Fact 10. Let $\mathcal{S} := \operatorname{convh}([V_i]_{i\in\mathbb{N}_{[1,q]}})$ and consider a positive scalar $a \in \mathbb{R}_+$. Then, $a\mathcal{S} := \operatorname{convh}(\{aV_i\}_{i\in\mathbb{N}_{[1,q]}})$.

Fact 11. Consider set $\mathcal{B}_1$ (20) and a vector $x \in \mathbb{R}^n$, $x \neq 0$. Then, $x \in \|x\|_1\mathcal{B}_1$.

Proof. Consider the matrix $V = [I_n \quad -I_n]$, which has as its columns the vertices of $\mathcal{B}_1$. For any $x \in \mathbb{R}^n$, $x \neq 0$, it holds that

$$\begin{aligned} x &= x^+ - x^- = V[x^{+\top} x^{-\top}]^\top \\ &= \|x\|_1 V[\|x\|_1^{-1} x^{+\top} \quad \|x\|_1^{-1} x^{-\top}]^\top \\ &= \|x\|_1 V\mu, \end{aligned}$$

where $\mu := \|x\|^{-1}[x^{+\top} \quad x^{-\top}]^\top$. Since $\mu \geq \mathbf{0}_{2n}$ and $\mathbf{1}_{2n}^\top \mu = \|x\|_1^{-1}\|x\|_1 = 1$, from Fact 10 we obtain $x \in \|x\|_1\mathcal{B}_1$. ■

4.2 Controller synthesis

A systematic way of computing offline the set–valued periodic vertex–interpolation control laws (8)–(10) can be formulated by characterizing the set $\mathcal{B}_1$ as a controlled (k, λ)–contractive set. The method is described in an algorithmic fashion below.

Algorithm 1 Controller synthesis problem.

Input: Matrices $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}$.
Output: k, λ, matrix sequences $\{U_i\}_{i\in\mathbb{N}_{[0,k-1]}}$,$\{V_i\}_{i\in\mathbb{N}_{[0,k]}}$.

01. **Assign** $k \leftarrow 0$, solution$\leftarrow 0$.
02. **while** solution$= 0$
03. solution$\leftarrow 1$
04. $k \leftarrow k + 1$
05. **for all** $j \in \mathbb{N}_{[1,n]}$
06. $v_0^j \leftarrow [I_n]_{:j}$
07. solve $\min_{\{u_i^j\}_{i\in\mathbb{N}_{[0,k-1]}}, \{v_i^j\}_{i\in\mathbb{N}_{[1,k]}}, p^j} \mathbf{1}_{2n}^\top p^j$
08. subject to
09. $v_{i+1}^j = Av_i^j + Bu_i^j$, $i \in \mathbb{N}_{[0,k-1]}$
10. $v_k^j = Vp^j, \quad p^j \geq \mathbf{0}_{2n}$
11. **if** $\mathbf{1}_{2n}^\top p^j \geq 1$
12. solution$\leftarrow 0$
13. **end**
14. **end**
15. **end**
16. $\lambda \leftarrow \max_{j\in\mathbb{N}_{[1,n]}} \mathbf{1}_{2n}^\top p^j$
17. **for all** $(i, j) \in \mathbb{N}_{[0,k-1]} \times \mathbb{N}_{[1,n]}$
18. $[U_i]_{:j} := u_i^j, \quad [U_i]_{:n+j} := -u_i^j$
19. $[V_i]_{:j} := v_i^j, \quad [V_i]_{:n+j} := -v_i^j$
20. **end**

Remark 12. Since $\mathcal{B}_1$ is a symmetric set, the optimization problem in Lines 07–10 of Algorithm 1 needs to be solved only n times, that correspond to the vertices of the set $\mathcal{B}_1$ that lie the positive orthant. Indeed, let the vector sequences $\{u_i^j\}_{i\in\mathbb{N}_{[0,k-1]}}, \{v_i^j\}_{i\in\mathbb{N}_{[1,k]}}$ denote the optimal solution of the problem, with $v_0^j = [V]_{:j}$, $i \in \mathbb{N}_{[1,n]}$ and consider the optimization problem in Lines 07–10 of Algorithm 1, with $v_0^{j+n} := [V]_{:j+n} = -[I_n]_{:j}$, $j \in \mathbb{N}_{[1,n]}$. Then, the optimization constraint in Line 09 is satisfied with $u_i^{j+n} := -u_i^j$, $i \in \mathbb{N}_{[0,k-1]}$, while the optimization constraints in Line 10 are satisfied with $[p^{j+n}]_i = [p^j]_{n+i}$, $i \in \mathbb{N}_{[1,n]}$, and $[p^{j+n}]_{n+i} = [p^j]_i$, $i \in \mathbb{N}_{[1,n]}$. Thus, for the particular selection of the set $\mathcal{B}_1$, the number of linear programs needed to be solved in the controller synthesis problem is equal to the state–space dimension.

Remark 13. The controller synthesis problem can also include hard state and input constraints, i.e., $x_t \in \mathbb{X} \subset \mathbb{R}^n$, $u_t \in \mathbb{U} \subset \mathbb{R}^m$, $\forall t \in \mathbb{N}$, with a small modification in the optimization problem in Lines 07–10 of Algorithm 1. In specific, two additional sets of constraints have to be added, i.e.,

$$v_i^j \in \mathbb{X}, \quad i \in \mathbb{N}_{[1,k]} \tag{21}$$

$$u_i^j \in \mathbb{U}, \quad i \in \mathbb{N}_{[0,k-1]}. \tag{22}$$

For the case when $\mathbb{X}, \mathbb{U}$ are proper $\mathcal{C}$–polytopic sets described in half–space or vertex representation, relations (21),(22) can be transformed to equivalent linear algebraic relations, as stated in Fact 1.

Although the controller synthesis problem is solved offline, the proposed algorithmic procedure can be found difficult to implement due to the large number of independent variables of the optimization problem in Lines 07–10 of Algorithm 1. For this reason, it is established next that the controller synthesis problem can be solved in a distributed fashion. In detail, we show that each linear program in Lines 07–10 of Algorithm 1 can be solved using distributed optimization methods. To this end, let the system (3) be decomposed in N subsystems, $N \in \mathbb{N}_{\leq n}$. For all $t \in \mathbb{N}$, consider the following decomposition of the state and input vector

$$x_t := \begin{bmatrix} x_{t,1}^\top & x_{t,2}^\top & \dots & x_{t,N}^\top \end{bmatrix}^\top, \tag{23}$$

$$u_t := \begin{bmatrix} u_{t,1}^\top & u_{t,2}^\top & \dots & u_{t,N}^\top \end{bmatrix}^\top, \tag{24}$$

where $x_{i,j} \in \mathbb{R}^{n_i}$, $(i,j) \in \mathbb{N} \times \mathbb{N}_{[1,N]}$ $u_{i,j} \in \mathbb{R}^{m_i}$, $(i,j) \in \mathbb{N} \times \mathbb{N}_{[1,N]}$, and $\sum_{i=1}^N n_i = n$, $\sum_{i=1}^N m_i = m$. The dynamics of each subsystem l, $l \in \mathbb{N}_{[1,N]}$, is shown below:

$$x_{t+1,l} = \sum_{j=1}^N A_{l,j} x_{t,j} + B_{l,j} u_{t,j}, \quad \forall (t,l) \in \mathbb{N} \times \mathbb{N}_{[1,N]}, \tag{25}$$

where matrices $A_{l,j} \in \mathbb{R}^{n_l \times n_j}$, $B_{l,j} \in \mathbb{R}^{n_l \times m_j}$ are the appropriately constructed blocks of matrices A, B, i.e.,

$$[A_{l,j}]_{ab} := [A]_{(\sum_{i=1}^{l-1} n_i + a)(\sum_{i=1}^{j-1} n_i + b)},$$

where $(a,b) \in \mathbb{N}_{[1,n_l]} \times \mathbb{N}_{[1,n_j]}$, and

$$[B_{l,j}]_{ab} := [B]_{(\sum_{i=1}^{l-1} n_i + a)(\sum_{i=1}^{j-1} m_i + b)},$$

where $(a,b) \in \mathbb{N}_{[1,n_l]} \times \mathbb{N}_{[1,m_j]}$.

The linear program in Lines 07–10 of Algorithm 1 can be reformulated in order to obtain decomposable optimization cost and constraints. In detail, the optimization constraint in Line 09 can be replaced, using the decompositions (23),(24) of the state and input vectors, by the following set of constraints

$$v_{i+1,l}^j = \sum_{r=1}^N \left(A_{l,r} v_{i,r}^j + B_{l,r} u_{i,r}^j \right), \forall (i,l) \in \mathbb{N}_{[0,k-1]} \times \mathbb{N}_{[1,N]}. \tag{26}$$

In addition, due to the shape of the set $\mathcal{B}_1$, the optimization constraint in Line 10 of Algorithm 1 can be also decomposed. Consider the matrices $V^i \in \mathbb{R}^{n_i \times 2n_i}$, $V^i := [I_{n_i} \quad -I_{n_i}]$, $i \in \mathbb{N}_{[1,N]}$. Then, the optimization constraint in Line 10 is equivalent to

$$v_{k,l}^j := V^l p_l^j, \quad p_l \geq 0, \quad \forall l \in \mathbb{N}_{[1,N]}, \tag{27}$$

where $p_l^j \in \mathbb{R}_+^{2n_l}$, $l \in \mathbb{N}_{[1,N]}$. Consequently, the optimization problem in Lines 07–10 in Algorithm 1 is equivalent to

$$\min_{\{u_{i,l}^j\}_{(i,l) \in \mathbb{N}_{[0,k-1]} \times \mathbb{N}_{[1,N]}}, \{v_{i,l}^j\}_{(i,l) \in \mathbb{N}_{[1,k]} \times \mathbb{N}_{[1,N]}}, \{p_l^j\}_{l \in \mathbb{N}_{[1,N]}}} \sum_{s=1}^N \mathbf{1}_{2n_i}^\top p_l^j \tag{28}$$

subject to (26),(27).

We are in a position to decompose the optimization problem (28),(26),(27) in N independent smaller problems which are coupled by equality [2] constraints.

In specific, let y_l, $l \in \mathbb{N}_{[1,N]}$, be an augmented vector that includes all the local variables $\{v_{i,l}^j\}_{i \in \mathbb{N}_{[1,k]}}$, $\{u_{i,l}^j\}_{i \in \mathbb{N}_{[0,k-1]}}$, p_l^j of subsystem l, as well as copies of the state and input sequences of the other subsstems that appear in (26). Also, let z denote the global variable, i.e. the augmented vector that contains all independent variables in Problem (28), (26)–(27). Let E_l, $l \in \mathbb{N}_{[1,N]}$, denote the matrices that assign the components of the local variables y_i to the global variable z, i.e.,

$$y_l := E_l z, \quad l \in \mathbb{N}_{[1,N]}.$$

Next, let $\mathcal{Y}_l$, $l \in \mathbb{N}_{[1,N]}$ denote the local constraint sets, defined by the relations (26),(27), which concern the local variable y_l. It is worth noting that sets $\mathcal{Y}_i$, $i \in \mathbb{N}_{[1,N]}$, are convex and in specific polytopic sets. Lastly, consider the functions,

$$f_l(y_l) := \mathbf{1}_{2n_l}^\top p_l^j, \quad l \in \mathbb{N}_{[1,N]},$$

which represent the local optimization costs. Then, Problem (28), (26)–(27) can be written as follows.

Problem 14. Solve the following optimization problem.

$$\min_{z, \{y_i\}_{i \in \mathbb{N}_{[1,N]}}} \sum_{i=1}^N f_i(y_i) \tag{29}$$

subject to

$$y_i \in \mathcal{Y}_i, \quad i \in \mathbb{N}_{[1,N]}, \tag{30}$$

$$y_i = E_i z, \quad i \in \mathbb{N}_{[1,N]}. \tag{31}$$

Several different distributed optimization methods, see for example (Boyd et al., 2011; Bertsekas and Tsitsiklis, 1989), can be exploited to solve Problem 14. These methods have a long history of being utilized in distributed control problems, see e.g. the most recent relevant works in distributed model predictive control (Summers and Lygeros, 2012; Conte et al., 2012). In specific, Problem 14 can be seen as a general form global consensus problem (Boyd et al., 2011). For this class of problems, the alternating direction method of multipliers can be utilized. To this end, consider the augmented Lagrangian

$$\begin{aligned} L(\{y_i\}_{i \in \mathbb{N}_{[1,N]}}, z, \delta) &= \sum_{i=1}^N f_i(y_i) \\ &\quad + \delta_i^\top (y_i - E_i z) + \frac{\rho}{2} \|y_i - E_i z\|_2^2, \\ &= \sum_{i=1}^N L_i(y_i, z, \delta_i), \end{aligned} \tag{32}$$

[2] In the general setting, matrices A, B in system (3) are full. Thus, for each subsystem dynamics, all state and input sequences of the other subsystems are appearing. However, in case a structure exists, as it is the case in interconnected large scale linear systems, the coupling variables are subsets of the overall global variable.

where δ is a vector containing the Lagrange multipliers, $\rho \in \mathbb{R}_+$ and the third term in the right hand equation can be seen an additional penalty term. Then, the alternating direction method of multipliers is realized by the iterations

$$\begin{aligned} y_i^{d+1} &= \text{argmin}_{y_i \in \mathcal{Y}_i} L_i(y_i, z^d, \delta_i^d) \\ z^{d+1} &= \text{argmin}_z L(\{y_i^{d+1}\}_{i \in \mathbb{N}_{[1,N]}}, z, \delta^d) \\ \delta_i^{d+1} &= \delta_i^d + \rho(y_i^{\delta+1} - E_i z^{d+1}), \end{aligned}$$

and it is initialized with $z := 0$, $\delta := 0$. Since the optimization cost and the constraint sets are linear functions, convergence to the optimal solution is guaranteed (Boyd et al., 2011). Problem 14 is jointly minimized over y_i, $i \in \mathbb{N}_{[1,N]}$ and z, with sequential minimizations between y_i, $i \in \mathbb{N}_{[1,N]}$ and z. This method allows the parallelization of minimizations of variables y_i, $i \in \mathbb{N}_{[1,N]}$. For the particular setting under study, the update of each component z_i reduces to local averaging of all solutions obtained by each local subsystem where the specific component appears.

4.3 Controller implementation

As shown in the previous section, a scalable and tractable construction of the control law (8)–(10) can be established by selecting as a controlled (k, λ)–contractive set the set $\mathcal{B}_1$. Then, the matrix sequence $\{U_i\}_{[i \in \mathbb{N}_{[0,k-1]}]}$, which defines the control law (8), is computed by applying Algorithm 1, or equivalently, by solving n linear programs. Furthermore, for large scale interconnected systems, a further decomposition of each linear program in Lines 07–10 of Algorithm 1 is possible by exploiting well established distributed optimization techniques. In this section, our goal is to establish a tractable online implementation of the set–induced periodic vertex–interpolation control law (8)–(10).

To this end, consider the control law

$$\pi(x_t) := U_i \overline{x}_{kN} \quad \text{if} \quad t = kN + i,\ N \in \mathbb{N}. \tag{33}$$

Theorem 15. Consider system (3), the controlled (k, λ)– contractive set $\mathcal{B}_1$ (20), the matrix sequence $\{U_i\}_{i \in \mathbb{N}_{[0,k-1]}}$, which is a solution of Algorithm 1, and the state–feedback control law (33). Then, the closed–loop linear system is globally $\mathcal{KL}$–stable.

Proof Sketch. From Facts 8,10,11, it follows that for any $x \in \mathbb{R}^n$, $x = \|x\|_1 V\mu$, where $\mu := \|x\|_1^{-1}\overline{x}$. Applying the control law (33) in system (3), it holds that

$$\begin{aligned} x_1 &= Ax + B\pi(x) = \|x\|_1 AV\mu + BU_0\overline{x} \\ &= \|x\|_1(AV\mu + BU_0\mu) = \|x\|_1 V_1\mu. \end{aligned}$$

Applying successively k times the control law (33) it follows that

$$x_k = \|x\|_1 V_k \mu.$$

From Algorithm 1 and Remark 12, there exist vectors $p^j \in \mathbb{R}_+^{2n}$, $j \in \mathbb{N}_{[1,2n]}$, such that $\mathbf{1}_{2n}^\top p^j < 1$, and $[V_k]_{:j} = Vp^j$, for all $i \in \mathbb{N}_{[1,2n]}$. Consider the matrix $P \in \mathbb{R}^{n \times 2n}$, where $[P]_{:j} = p^j$, $j \in \mathbb{N}_{[1,2n]}$. Then, $x_k = \|x\|_1 V_k\mu = \|x\|_1 VP_k\mu$, and since $\mathbf{1}_{2n}^T P_k \mu \leq \lambda$, from Fact 1 it follows that $x_k \in \lambda\|x\|_1\mathcal{B}_1$. Thus, the set $\mathcal{B}_1$ is (k, λ)–contractive with respect to the closed–loop system. The result follows from exploiting (Lazar et al., 2013, Theorem IV.5). The details are omitted here due to the space limitations. ■

The implications of the result are significant, since the resulting stabilizing law is explicit and does not require the solution of an online optimization problem, or a point location problem, as it is the case for vertex–interpolation control laws in general.

Consequently, the control law (33) can be implemented in a distributed manner. In specific, consider the decomposition of matrices U_i, $i \in \mathbb{N}_{[0,k-1]}$ in

$$U_i := \begin{bmatrix} U_i^+ & U_i^- \end{bmatrix},$$

where $U_i^+ \in \mathbb{R}^{m \times n}$, $i \in \mathbb{N}_{[0,k-1]}$, $U_i^- \in \mathbb{R}^{m \times n}$, $i \in \mathbb{N}_{[0,k-1]}$. Then, the j–th element $\pi_j(x_t) \in \mathbb{R}$, $j \in \mathbb{N}_{[1,m]}$ of the control law (33) can be written in the following form

$$\begin{aligned} \pi_j(x_t) &:= \sum_{c=1}^{n} \left([U_i^+]_{jc} x_c^+ + [U_i^-]_{jc} x_c^-\right) \\ &= \sum_{c=1}^{n} \pi_j^c(x_c), \forall j \in \mathbb{N}_{[1,m]},\ \text{if } t = kM + i, M \in \mathbb{N} \end{aligned} \tag{34}$$

where

$$\pi_j^c(x_c) := [U_i^+]_{jc} x_c^+ + [U_i^-]_{jc} x_c^-,\ \text{if } t = kM + i, M \in \mathbb{N}. \tag{35}$$

Thus, the control law (33) can be realized in three simple steps. First, given $k \in \mathbb{N}_{\geq 1}$, for any time instant t, the integers $M \in \mathbb{N}$ and $i \in \mathbb{N}_{[0,k-1]}$ are computed such that $t = kM + i$. Second, the values of the functions $\pi_j^c(x_t)$, $(j, c) \in \mathbb{N}_{[1,m]} \times N_{[1,n]}$, defined in (35), are computed, by performing a single multiplication for each term. Lastly, the control law $\pi_j(x_t)$ is computed for each $j \in \mathbb{N}_{[1,m]}$, by performing an addition of the terms $\pi_j^c(x_t)$ calculated at the second step.

Remark 16. The control law (33) is positively homogeneous of the first degree. Since the linear system (3) is homogeneous of the first degree with respect to both arguments, the resulting closed–loop system is also homogeneous. From (Lazar et al., 2013, Corollary V.3), it follows that for this class of systems, global $\mathcal{KL}$–stability equals global exponential stability. Thus, the closed–loop system under the state–feedback control law (33) is globally exponentially stable.

5. ILLUSTRATIVE EXAMPLES

In this section, two illustrative examples are presented.

5.1 Example 1

We consider a linear large scale system (3) with $n = 1000$, $m = 100$. The system has 180 unstable eigenvalues and it is not controllable. However, it is stabilizable. The eigenvalues of matrix A are shown in the complex plane in Figure 2. The matrices $A \in \mathbb{R}^{1000 \times 1000}$ and $B \in \mathbb{R}^{1000 \times 100}$ are of full structure. By applying Algorithm 1, the set $\mathcal{B}_1$ was characterized as a controlled (k, λ)–contractive set for $k = 10$, $\lambda = 0.88$. The linear program in Lines 07–10 of Algorithm 1 was solved for the nonnegative 1000 vertices of $\mathcal{B}_1$, using the IBM ILOG CPLEX Optimizer in an up-to-date standard desktop computer. The average time spent for solving each optimization problem was 5 seconds. In Figure 3, the state response of the closed–loop system with control law (33) is shown for the initial condition $x_0 \in \mathbb{R}^{1000}$, where $[x_0]_1 := 10$, $[x_0]_i := 0$, $i \in \mathbb{N}_{[1,1000]} \setminus \{1\}$, while the corresponding control effort is shown in Figure 4. The average online computation time for different initial conditions was found to be negligible (under 0.1ms), compared to the computation time needed to solve the offline controller synthesis problem.

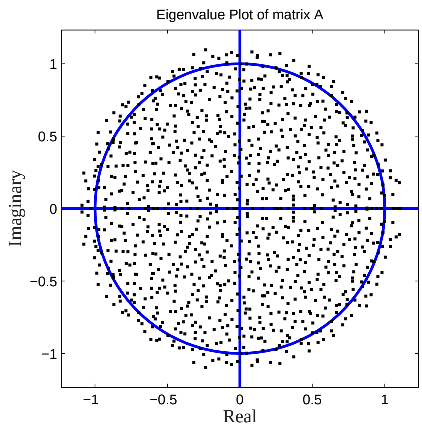

Fig. 2. The eigenvalues of matrix A in the complex plane, Example 1.

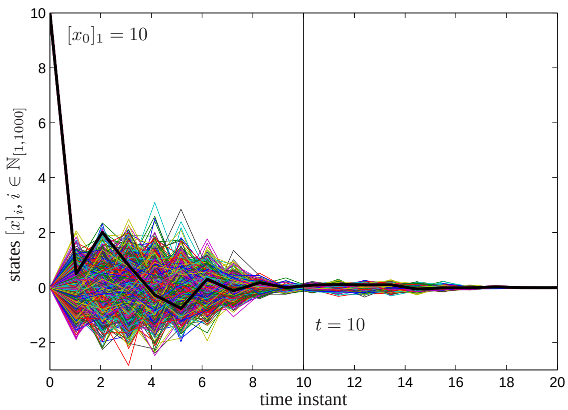

Fig. 3. State response of the closed–loop system under the global periodic vertex interpolation control law (33), for the initial condition $[x_0]_1 = 10$, $[x_0]_i = 0$, $i \in \mathbb{N}_{[1,1000]} \setminus \{1\}$, Example 1.

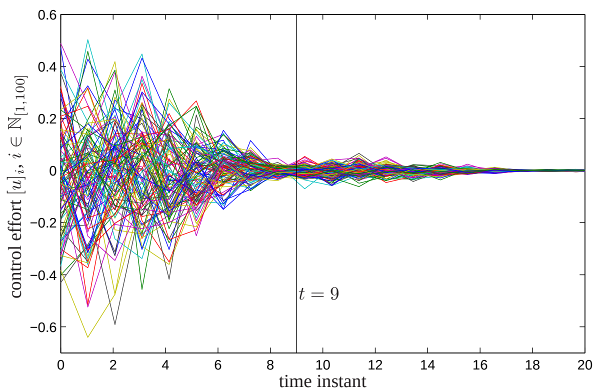

Fig. 4. Control effort for the initial condition $[x_0]_1 = 10$, $[x_0]_i = 0$, $i \in \mathbb{N}_{[1,1000]} \setminus \{1\}$, Example 1.

5.2 *Example 2*

We consider a linear large scale system (3) with $n = 1000$, $m = 100$. The system has 50 unstable eigenvalues and it is controllable. The matrices $A \in \mathbb{R}^{1000\times 1000}$ and $B \in \mathbb{R}^{1000\times 100}$ have a specific structure, which is shown in Figure 5.

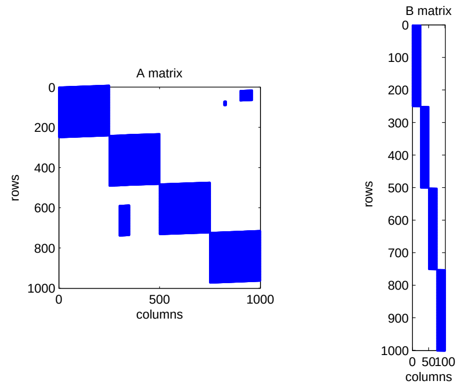

Fig. 5. The sparsity pattern of matrices A, B (the blue color denotes a nonnegative element), Example 2.

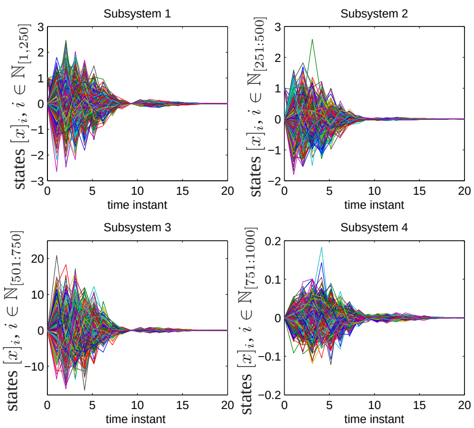

Fig. 6. State response of the closed–loop system under the global periodic vertex interpolation control law (33), for the initial condition $[x_0]_i = 1$, $i \in \mathbb{N}_{[50,100]} \cup \mathbb{N}_{[350:400]}$, $[x_0]_i = 0$, $i \in \mathbb{N}_{[1,49]} \cup \mathbb{N}_{[101:349]} \cup \mathbb{N}_{[401:1000]}$, Example 2.

From the sparsity pattern of the matrix pair (A, B), we can consider the system as an interconnected system consisting of $N = 4$ subsystems which have 250 state variables and 25 control inputs. Due to this specific structure, Algorithm 1 can be implemented in a distributed fashion, utilizing the results in Section 4.2. Thus, the optimization problem in Lines 07–10 of Algorithm 1 can be brought to the form of Problem 14 and solved in a distributed fashion using the alternating direction method of multipliers, for each positive vertex of the set $\mathcal{B}_1$. Nevertheless, for comparison purposes with Example 1, the global linear program in Lines 07–10 of Algorithm 1 was solved for the nonnegative 1000 vertices of $\mathcal{B}_1$, using the IBM ILOG CPLEX Optimizer in an up-to-date standard desktop computer. The average time spent for the convergence of the distributed optimization scheme was 1 second. It is worth noting that, even when solving the global problem, the computation time is significant lower than in Example 1 where the system under study was unstructured. This observation indicates that the method proposed inherently exploits the structure of the system. Lastly, in Figure 6, the state response of the closed–loop system with control law (33) is shown for the initial condition for the initial condition $[x_0]_i = 1$, $i \in \mathbb{N}_{[50,100]} \cup \mathbb{N}_{[350:400]}$, $[x_0]_i = 0$, $i \in \mathbb{N}_{[1,49]} \cup \mathbb{N}_{[101:349]} \cup \mathbb{N}_{[401:1000]}$. Similarly to the previous example, the average online computation time for different initial conditions was found to be negligible (under 0.1ms), compared to the computation time needed to solve the offline controller synthesis problem.

6. CONCLUSIONS

A non–conservative and scalable synthesis method for large scale linear systems was presented. By exploiting the finite–time CLF concept, Minkowski functions of a particular family of polytopic sets, which includes the 1–norm hyper–rhombus, were selected as candidate finite–time CLFs. The selection of the unit sublevel set of the 1–norm as candidate controlled (k, λ)–contractive set resulted in globally stabilizing periodic vertex–interpolation control laws. A systematic method of constructing offline the vertex–control laws using distributed optimization in a scalable fashion was established, while it was shown that the actual control law is of an explicit, distributed form.

REFERENCES

Athanasopoulos, N., Doban, A.I., and Lazar, M. (2013). On Constrained Stabilization of Discrete–Time Linear Systems. In *21st Mediterranean Conference on Control and Automation (MED)*, 831–839. Chania, Greece.

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Prentice–Hall.

Boyd, S., Parikh, E., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3, 1–122.

Conte, C., Voellmy, N.R., Zeilinger, M.N., Morari, M., and Jones, C. (2012). Distributed Synthesis and Control of Constrained Linear Systems. In *American Control Conference*, 6017–6022. Montreal, Canada.

Dekker, G., Jokic, A., and Weiland, S. (2010). Distributed $\mathcal{H}_\infty$–based control of Electrical Power Systems. In *2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 19–24. Annecy, France.

Grujic, L.T. and Siljak, D.D. (1973). Asymptotic Stability and Instability of Large–Scale Systems. *IEEE Transactions on Automatic Control*, 636–645.

Gutman, P.O. and Cwikel, M. (1986). Admissble sets and feedback control for discrete-time linear systems with bounded control and states. *IEEE Transactions on Automatic Control*, AC-31(4), 373–376.

Johansson, B., Speranzon, A., Johansson, M., and Johansson, K. (2006). Distributed model predictive consensus. In *17th Symposium on Mathematical Theory of Networks and Systems*, 2438–2444. Kyoto, Japan.

Langbort, C., Chandra, R.M., and D'Andrea, R. (2004). Distributed Control Design for Systems Interconnected OVer an Arbitrary Graph. *IEEE Transaction on Automatic Control*, 49, 1502–1519.

Lazar, M., Raković, S.V., Doban, A.I., and Athanasopoulos, N. (2013). On stability analysis of discrete–time homogeneous dynamics. Technical Report, Eindhoven University of Technology, available at: `www.cs.ele.tue.nl/MLazar/TechReport03.pdf` .

Michel, A.N. and Miller, R.K. (1977). *Qualitative analysis of large scale dynamical systems*, volume 134 of *Mathematics in Science and Engineering*. Academic Press, Boston.

Rantzer, A. (2009). Dynamic Dual Decomposition for Distributed Control. In *American Control Conference*, 884–888. St. Louis, MO, USA.

Siljak, D.D. (1991). *Decentralized control of complex systems*. Academic Press, Boston.

Siljak, D.D. (2007). *Large-scale dynamic systems: Stability and structure*. Dover Publications. Reprint of the 1978 edition.

Summers, T.H. and Lygeros, J. (2012). Distributed Model Predictive Consensus via the Alternating Direction Method of Multipliers. In *Allerton Conference on Communication, Control and Computing*. Monticello, IL, USA. Available at `http://arxiv.org/pdf/1212.1296v1.pdf`.

Vidyasagar, M. (1981). *Input–Output Analysis of Large–scale Interconnected Systems*, volume 29 of *Lecture Notes in Control and Information Sciences*. Berlin, Germany: Springer–Verlag.

Ziegler, G.M. (2007). *Lectures on Polytopes, Updated Seventh Printing*. Springer.